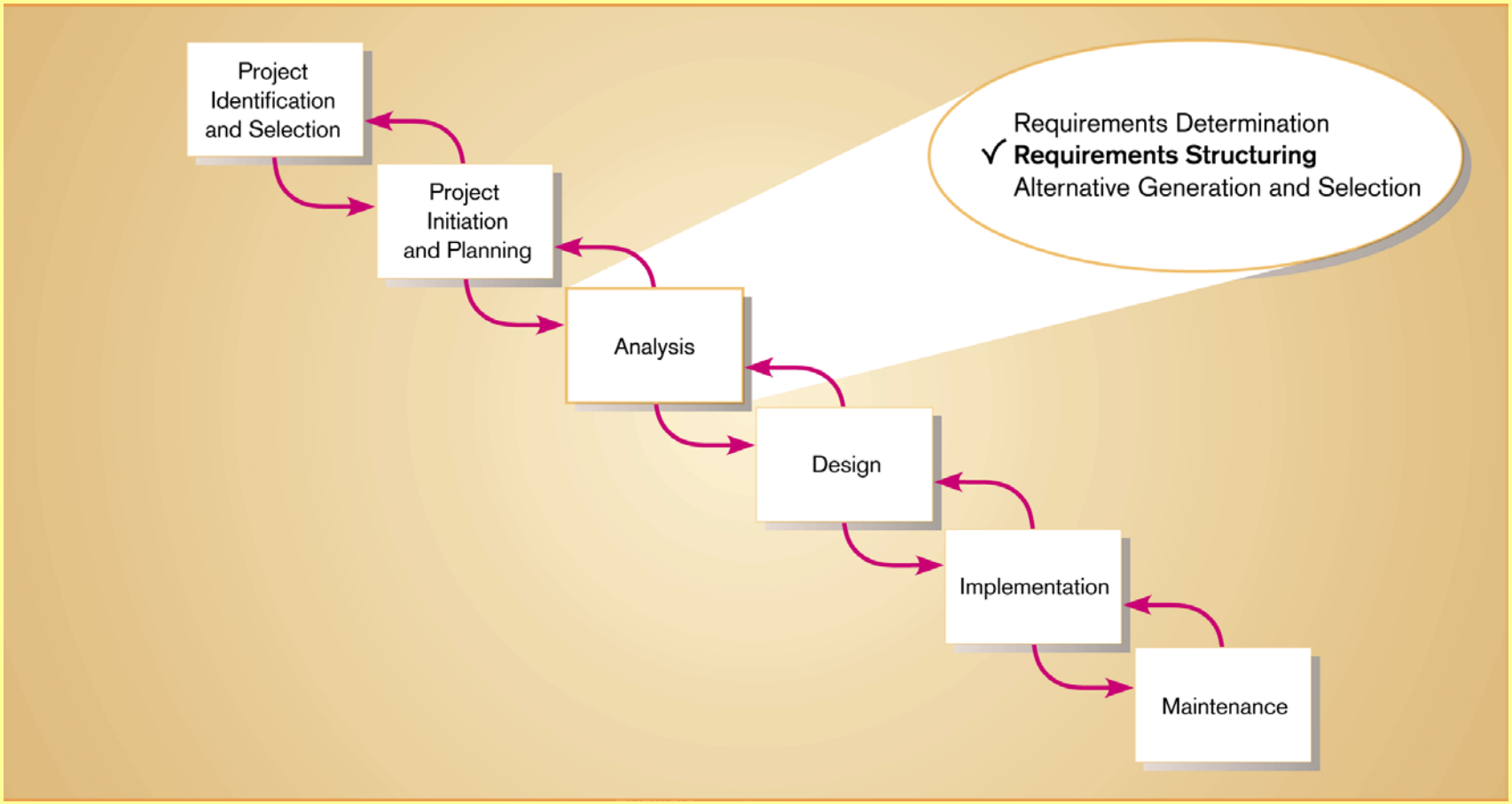
Chapter 5

Structuring System Process Requirements

In this chapter, we will concentrate on how to represent the information about processes gathered in the previous subphase.



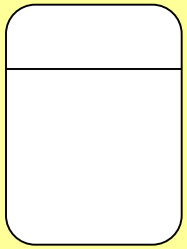
Process Modeling

- **Graphically represent the processes that: capture, manipulate, store and distribute data between a system and its environment and among system components.**
- **Data flow diagrams (DFD)**
  - **Graphically illustrate movement of data between external entities and the processes and data stores within a system**
- **Modeling a system's process**
  - **Utilize information gathered during requirements determination**
  - **Structure of the data is also modeled in addition to the processes**

Process Modeling

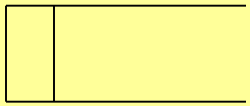
- **Deliverables and Outcomes**
- **Set of coherent, interrelated data flow diagrams containing:**
  - **Context data flow diagram (DFD)**
    - **Scope of system (what inside and outside the system)**
  - **DFDs of current physical system**
    - **Enables analysts to understand current system**
  - **DFDs of new and current logical system**
    - **Technology independent**
    - **Show data flows, structure and functional requirements of new system**

DFD Symbols



Process

Source/Destination



Data Store

Data Flow

1) Data Flow

- **Depicts data that are in motion and moving as a unit from one place to another in the system.**
- **Drawn as an arrow with the dataflow name on it.**

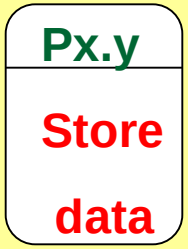
- **Its direction coming from the place that produces the data to the place that will use (store) the data.**
- **Select a meaningful name to represent the data**
- **Its name should be a noun (Employee data, Username, Food record, Add request)**
- **May be composed of many individual pieces of data (Username & Password, Employee name & Employee ID)**

2)Data Store

- **Depicts data at rest**
- **May represent data in**
  - **Physical File folder or Notebook**
  - **Computer-based file (table(s) of database, figures, plain text file,….)**
- **Choose a suitable name for the data store which should be a noun (Customers data, Students data, Images)**
- **Each data store is also given a code name (D1) which should not be repeated for another data store (and this is applied also for its name) as in Fig.**

3) Process

- **Depicts work or action or task performed on data so that they are transformed, stored or distributed in the system**
- **Choose a suitable name for the process which should be a verb (Update data, Store data, Find, Calculate monthly payment)**
- **Each process is given a number (or code) which should not be repeated for another process (and this is applied also for its name) as in Fig.**



4) Source/Sink

- **Depicts the origin and/or destination of the data**
- **Sometimes referred to as an external entity as they are outside the system we are studying.**
- **We get external data from a source.**
- **Once processed, data leaves the system and goes outside to a sink.**
- **They could be :**

1) another organization or a unit of it that sends or receives data from the system (supplier dept. or academic dept.)

2) a person inside or outside the system that interact with the system (a customer or loan offer or a librarian)

3) another information system that exchange information with our system.

- **Drawn as a square symbol.**

4) Source/Sink (continued)

- **We may have more than one source/sink in the system.**
- **Choose a suitable name that states what the external agent is (Bank, Customer, Employment system, Teller). It should be a noun.**
- **Because they are external, many characteristics are not of interest to us**
- **We do not care what they will do with information or how they operate (if they are another system producing or using the data).**
- **They can not access the data stores, so a process must be put between the source/sink and data store.**

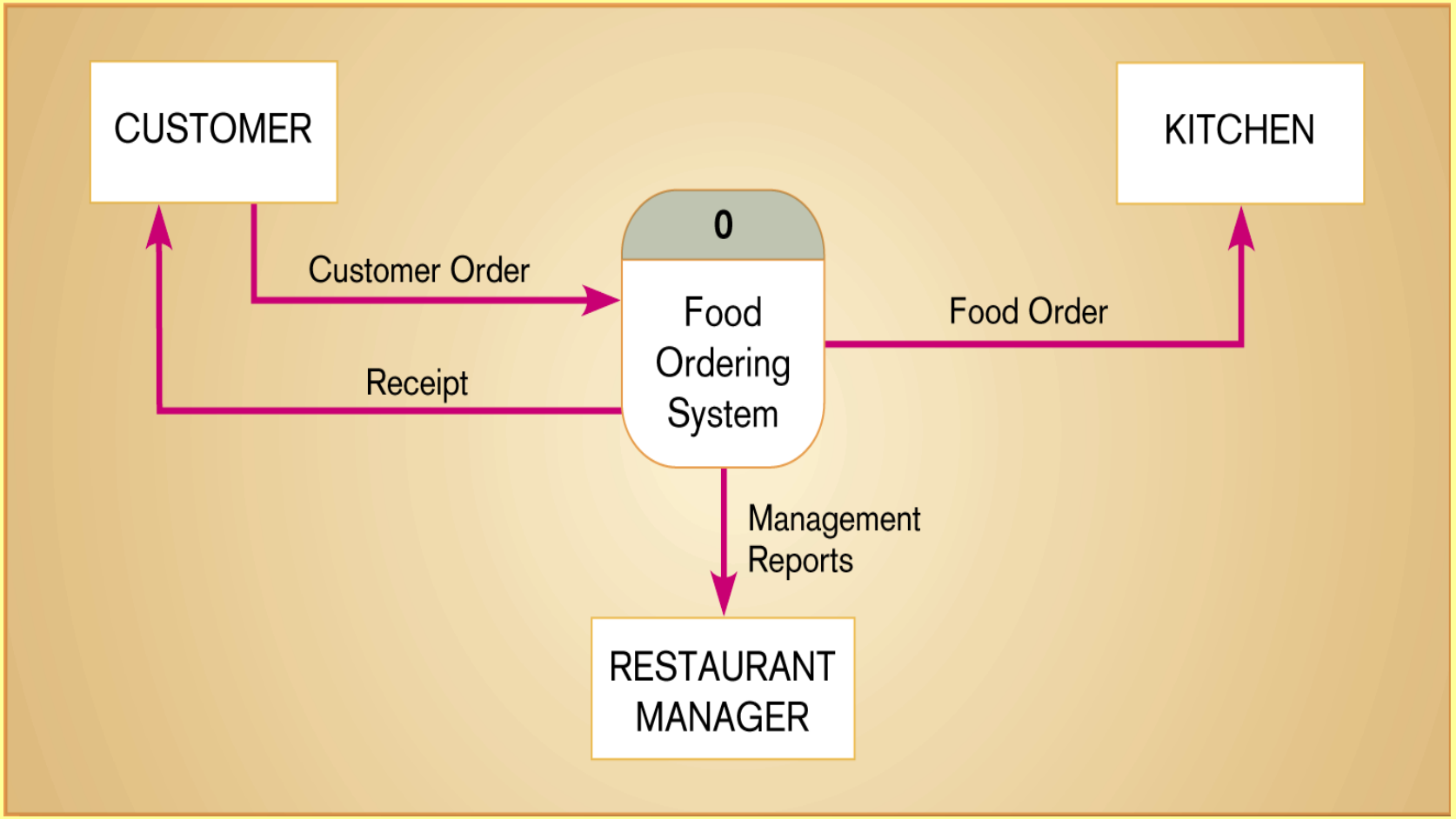
DFD Definitions

- **Context Diagram**
  - **A data flow diagram (DFD) of the scope of an organizational system that shows the system boundaries, external entities that interact with the system and the major information flows between the entities and the system**
- **Level-0 Diagram**
  - **A data flow diagram (DFD) that represents a system's major processes, data flows and data stores at a high level of detail**

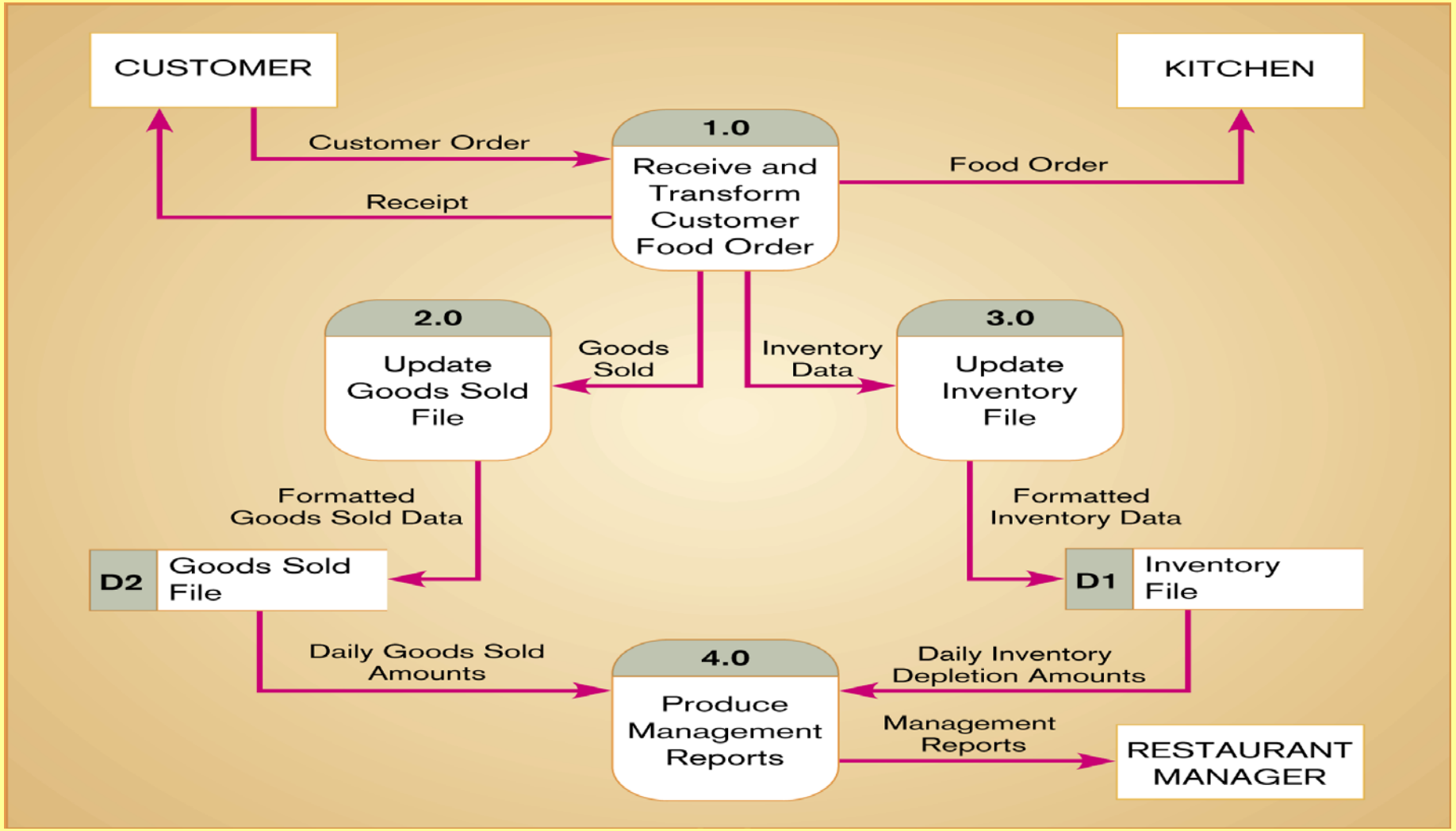
Developing DFDs: An Example

- **Hoosier Burger's automated food ordering system**
- **Context Diagram contains no data stores**
- **Next step is to expand the context diagram to show the breakdown of processes**

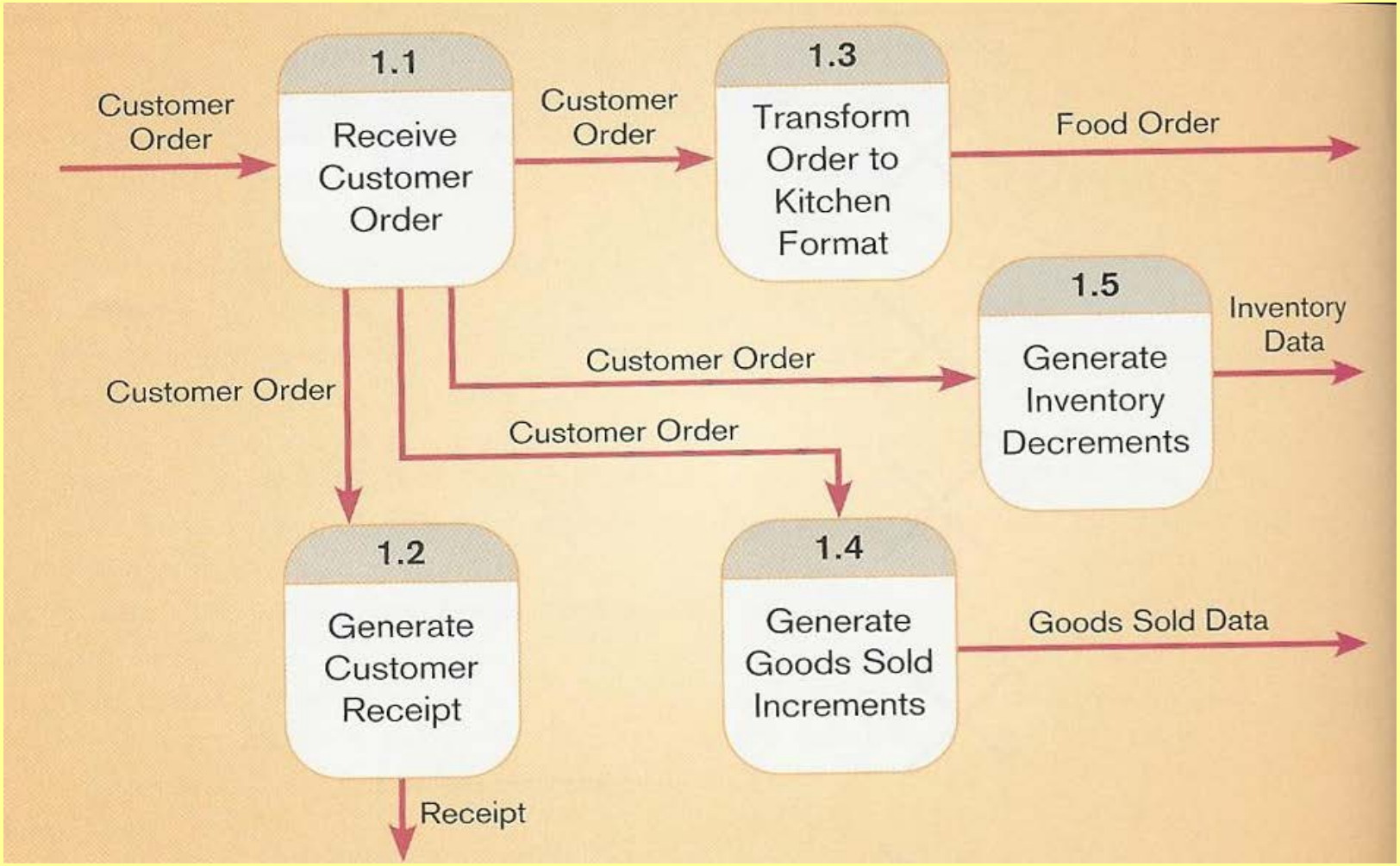
Context diagram of Hoosier Burger's food ordering system



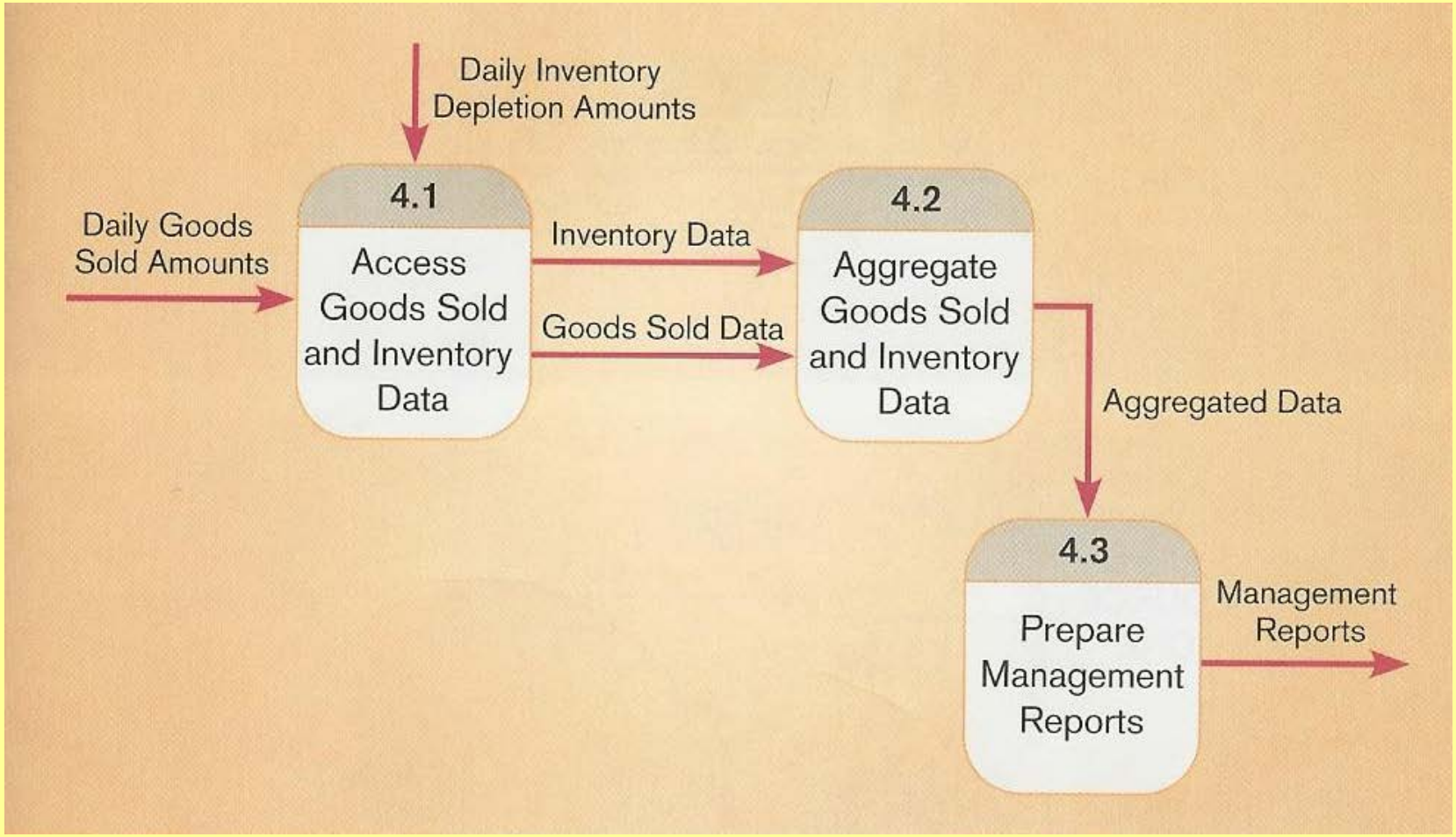
Level-0 DFD of Hoosier Burger's food ordering system



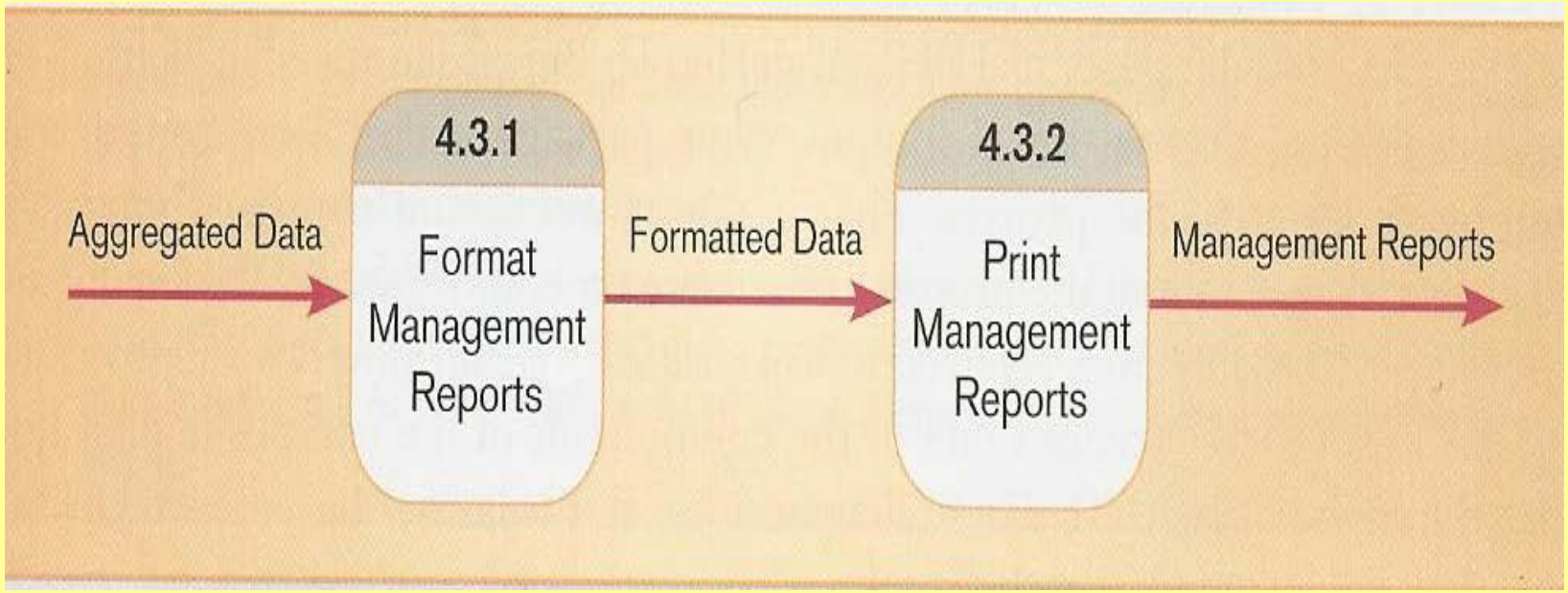
Level-1 DFD of Receive & Transform Customer Food Order



Level-1 DFD of Produce Management Report



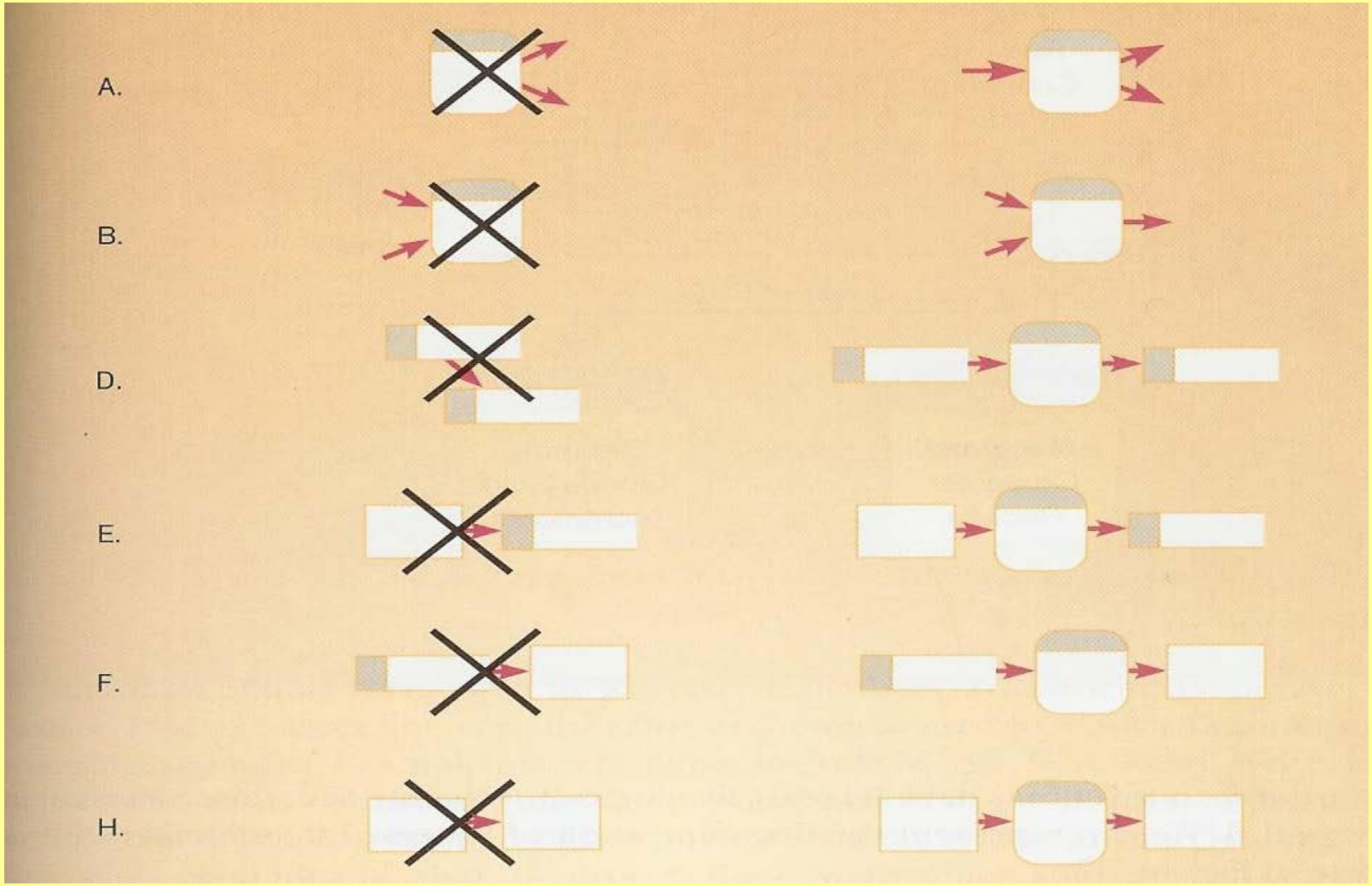
Level-2 DFD of Prepare Management Report



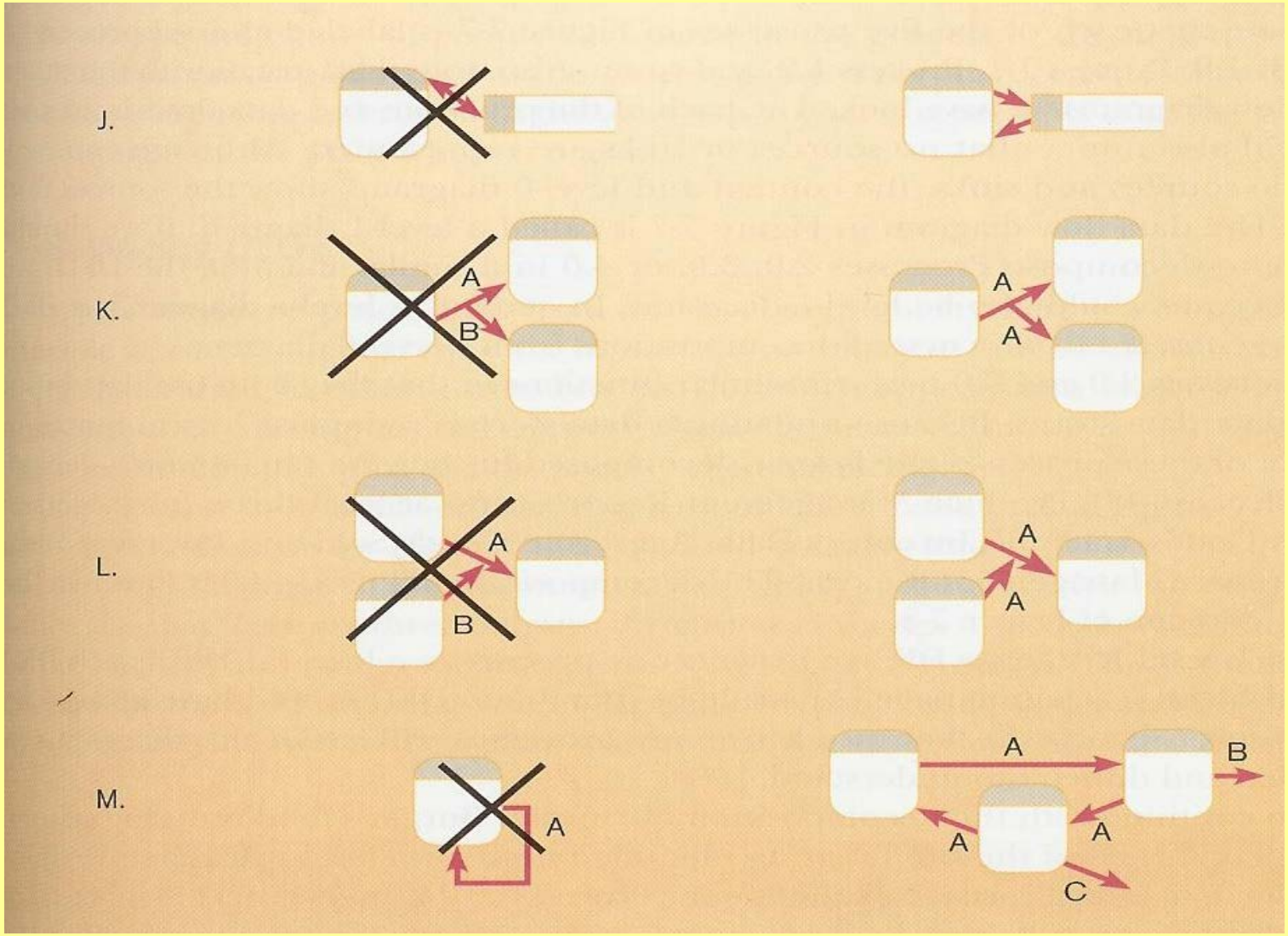
Data Flow Diagramming Rules

- **Basic rules that apply to all DFDs**
  - **Inputs to a process are mostly different than outputs**
  - **Objects always have a unique name**
    - **In order to keep the diagram uncluttered, you can repeat data stores and sources/sinks on a diagram**

DFD Rules



DFD Rules



DFD Rules

- Process
  - No process can have only outputs (a miracle)
  - No process can have only inputs (black hole)
  - A process has a verb phrase label

- **Data Store**
  - **Data cannot be moved directly from one store to another**
  - **Data cannot move directly from an outside source to a data store**
  - **Data cannot move directly from a data store to a data sink**
  - **Data store has a noun phrase label**

DFD Rules

- Source/Sink
  - Data cannot move directly from a source to a sink
  - A source/sink has a noun phrase label

- **Data Flow**
  - **A data flow has only one direction of flow between symbols**
  - **A fork means that exactly the same data goes from a common location to two or more processes, data stores or sources/sinks**

DFD Rules

- Data Flow (Continued)
  - L. A join means that exactly the same data comes from any two or more different processes, data stores or sources/sinks to a common location
  - M. A data flow cannot go directly back to the same process it leaves
  - N. A data flow to a data store means update
  - O. A data flow from a data store means retrieve or use
  - P. A data flow has a noun phrase label

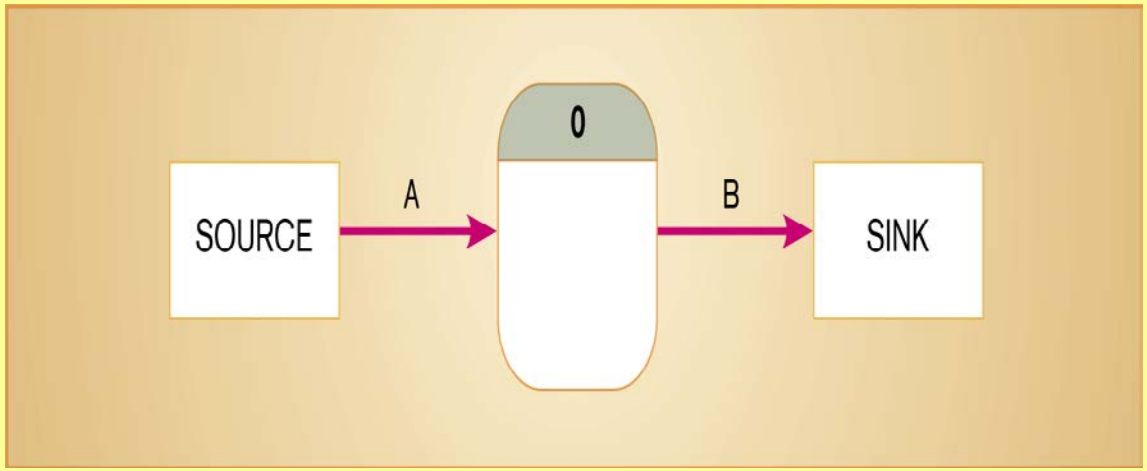
Decomposition of DFDs

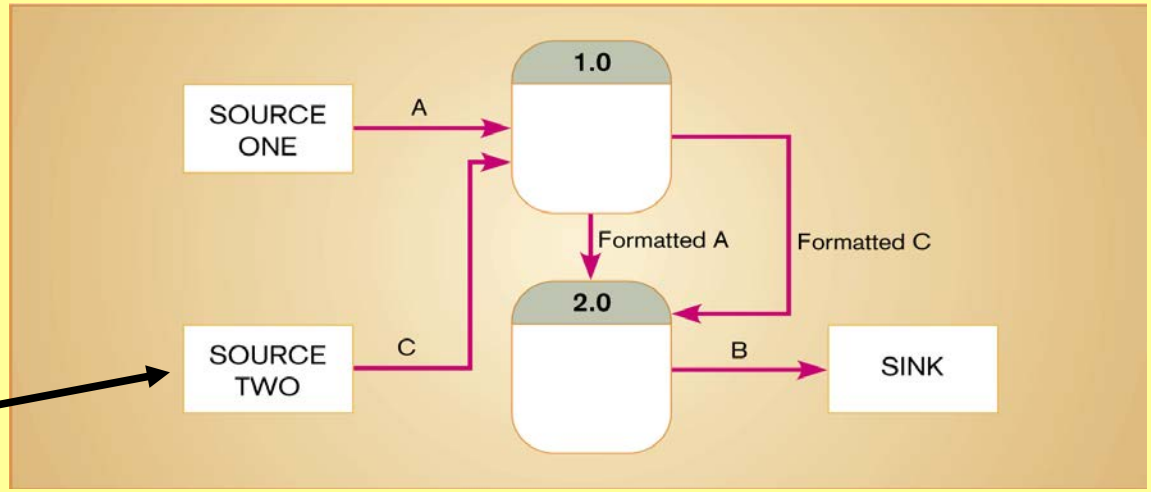
- **Functional decomposition**
  - We always begin with high level diagrams then decompose each large process in it into finer and simpler ones.
  - The act of going from one single system to many component processes is called functional decomposition.
  - It is a repetitive procedure creating a hierarchy of DFDs
  - One process on a given DFD is explained in greater details on another DFD.
  - The lowest level is called a primitive DFD
- **Level-N Diagram**
  - A DFD that is the result of *n* nested decompositions of a series of subprocesses from a process on a level-0 diagram

Balancing DFDs

- When decomposing a DFD, you must conserve inputs to and outputs from a process at the next level of decomposition
- I.e. The process must have the same inputs and outputs when it is decomposed to next level.
- This conservation is called balancing.
- Next fig. shows unbalanced DFD as in level-0 there is an additional source was added.

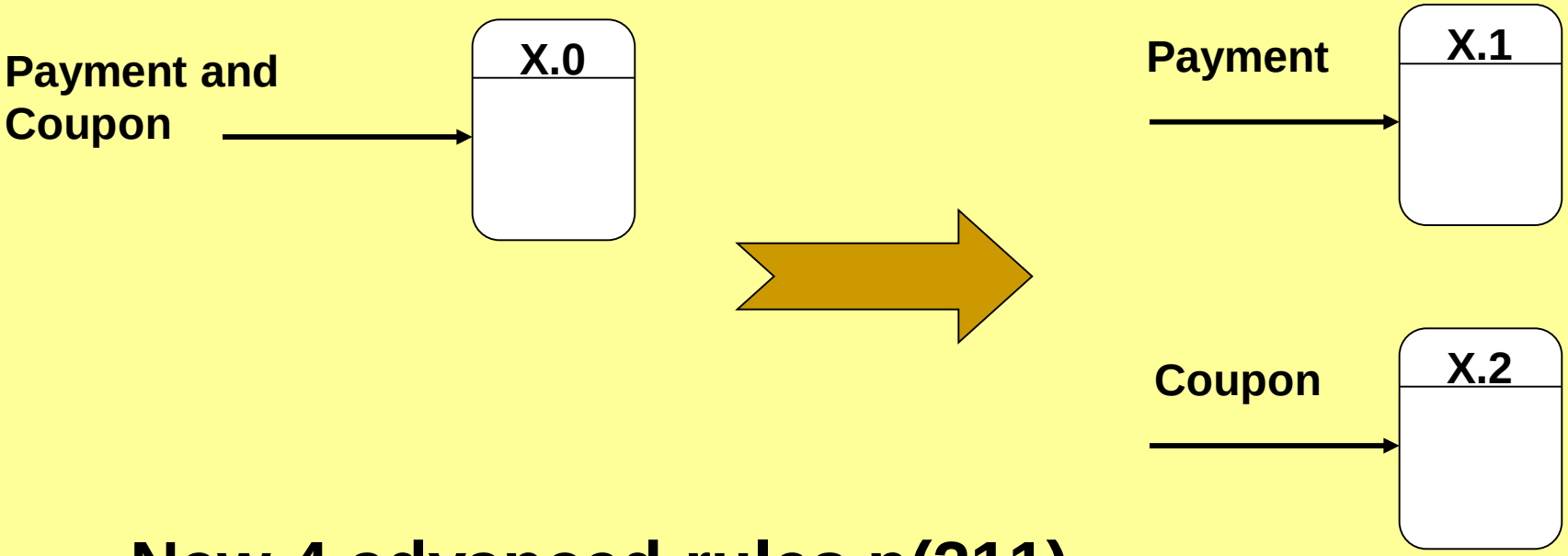
Unbalanced DFD





Balancing DFDs

- However, we can split a data flow into separate data flows on a lower level diagram

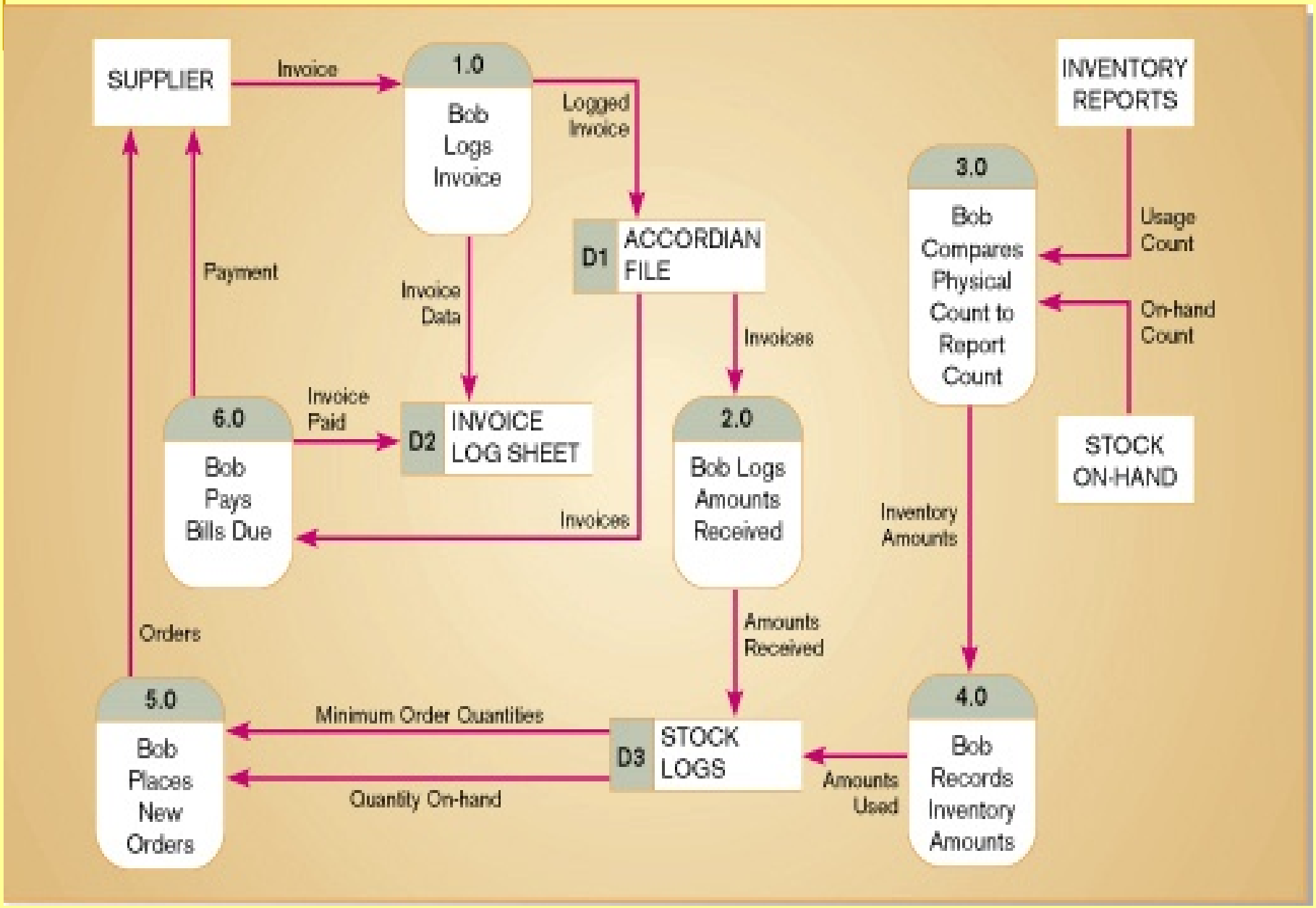


- New 4 advanced rules p(211)

Four Different Types of DFDS

1- Current Physical

- Process label includes an identification of the technology (people or systems) used to process the data
- Data flows and data stores are labeled with the actual name of the physical media on which data flow or in which data are stored (file folders, computer files)


SUPPLIER
Invoice
1.0
Bob Logs Invoice
Logged Invoice
INVENTORY REPORTS
3.0
Bob Compares Physical Count to Report Count
Usage Count
On-hand Count
Payment
Invoice Data
D1 ACCORDIAN FILE
Invoices
6.0
Bob Pays Bills Due
Invoice Paid
D2 INVOICE LOG SHEET
2.0
Bob Logs Amounts Received
STOCK ON-HAND
Invoices
Inventory Amounts
Orders
Amounts Received
5.0
Bob Places New Orders
Minimum Order Quantities
D3 STOCK LOGS
4.0
Bob Records Inventory Amounts
Quantity On-hand
Amounts Used

Four Different Types of DFDS

2- Current Logical

- Physical aspects of system are removed as much as possible
- Current system is reduced to data and processes that transform them

3- New Logical

- Is the same as the current one if the user is happy with the functionality of the current system.
- Includes additional functions
- Obsolete functions are removed
- Inefficient data flows are reorganized

Four Different Types of DFDS

4- New Physical

- Represents the physical implementation of the new system
- The analyst will\make decision on which parts of the system will be automated and which parts will be kept manual

Guidelines for Drawing DFDs

- **Completeness**
  - **DFD must include all components necessary for system**
  - **Each component must be fully described in the project dictionary**
- **Consistency**
  - **The extent to which information contained on one level of a set of nested DFDs is also included on other levels (ex. A data flow may appear in one level and not in the next level or with different name)**

Guidelines for Drawing DFDs

- **Timing**
  - **Time is not represented well on DFDs**
  - **Best to draw DFDs as if the system has never started and will never stop.**

- **Iterative Development**
  - **Analyst should expect to redraw diagram several times before reaching the closest approximation to the system being modeled**

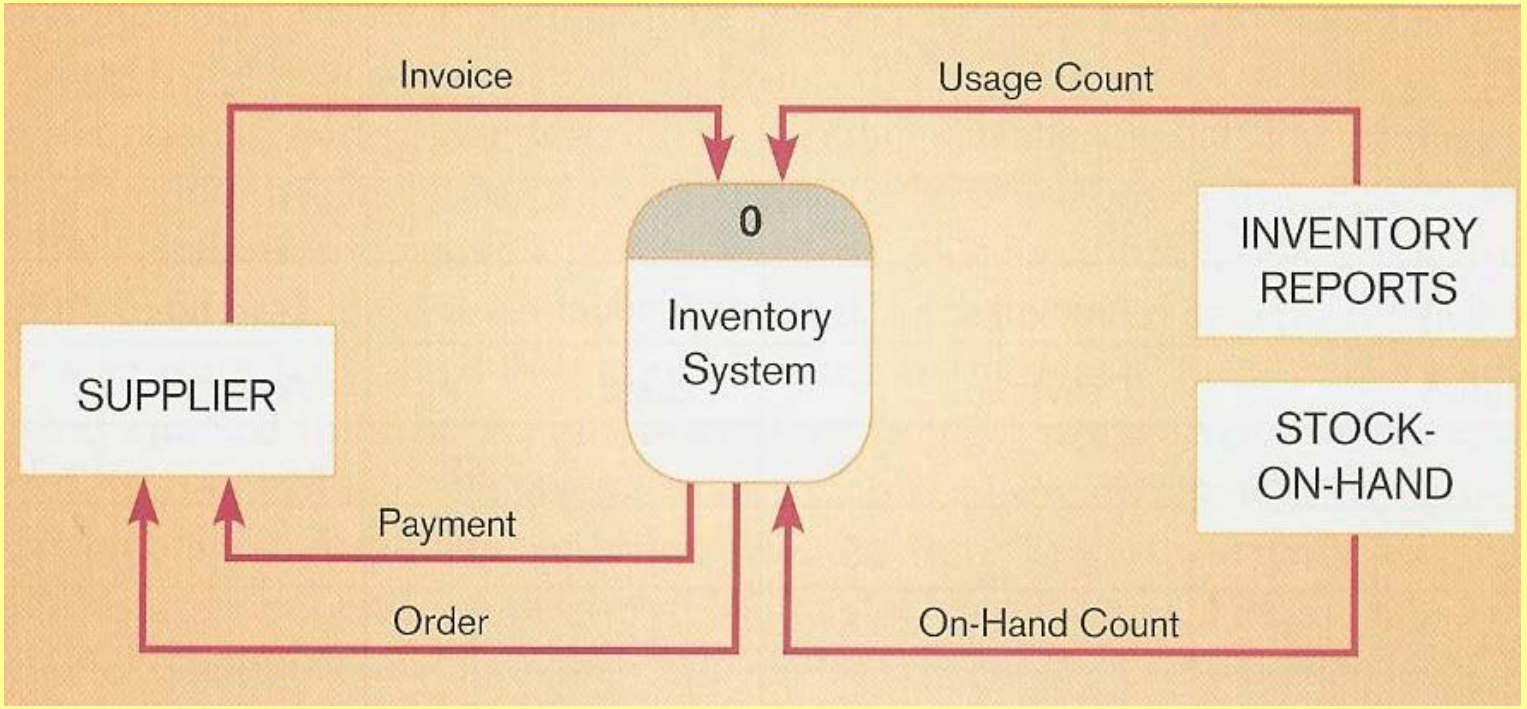
Guidelines for Drawing DFDs

- **Primitive DFDs**
  - **Lowest logical level of decomposition**
  - **Decision has to be made when to stop decomposition**

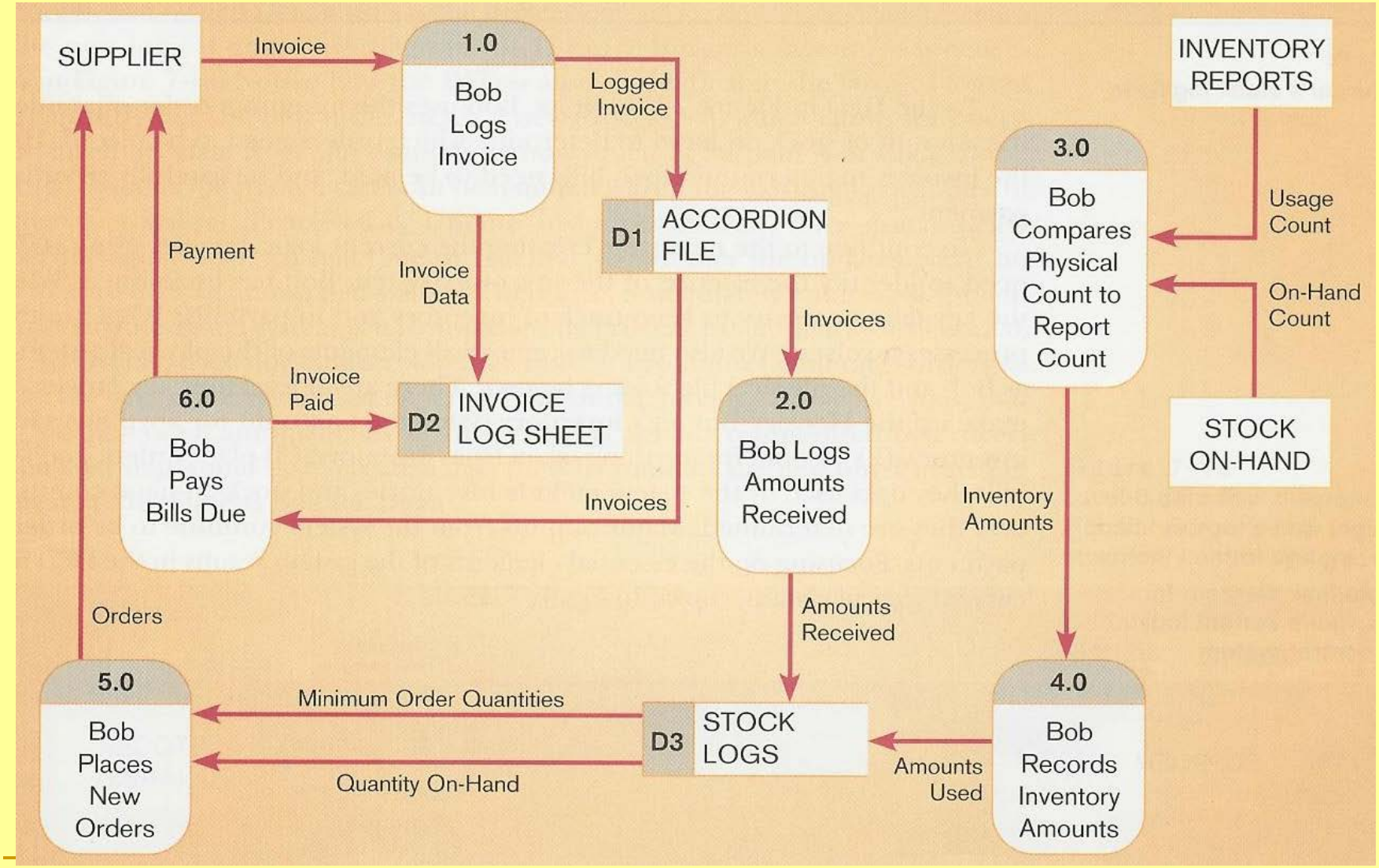
Rules for stopping decomposition

- **When each process has been reduced to a single decision, calculation or database operation**
- **When each data store represents data about a single entity**
- **When the system user does not care to see any more detail**

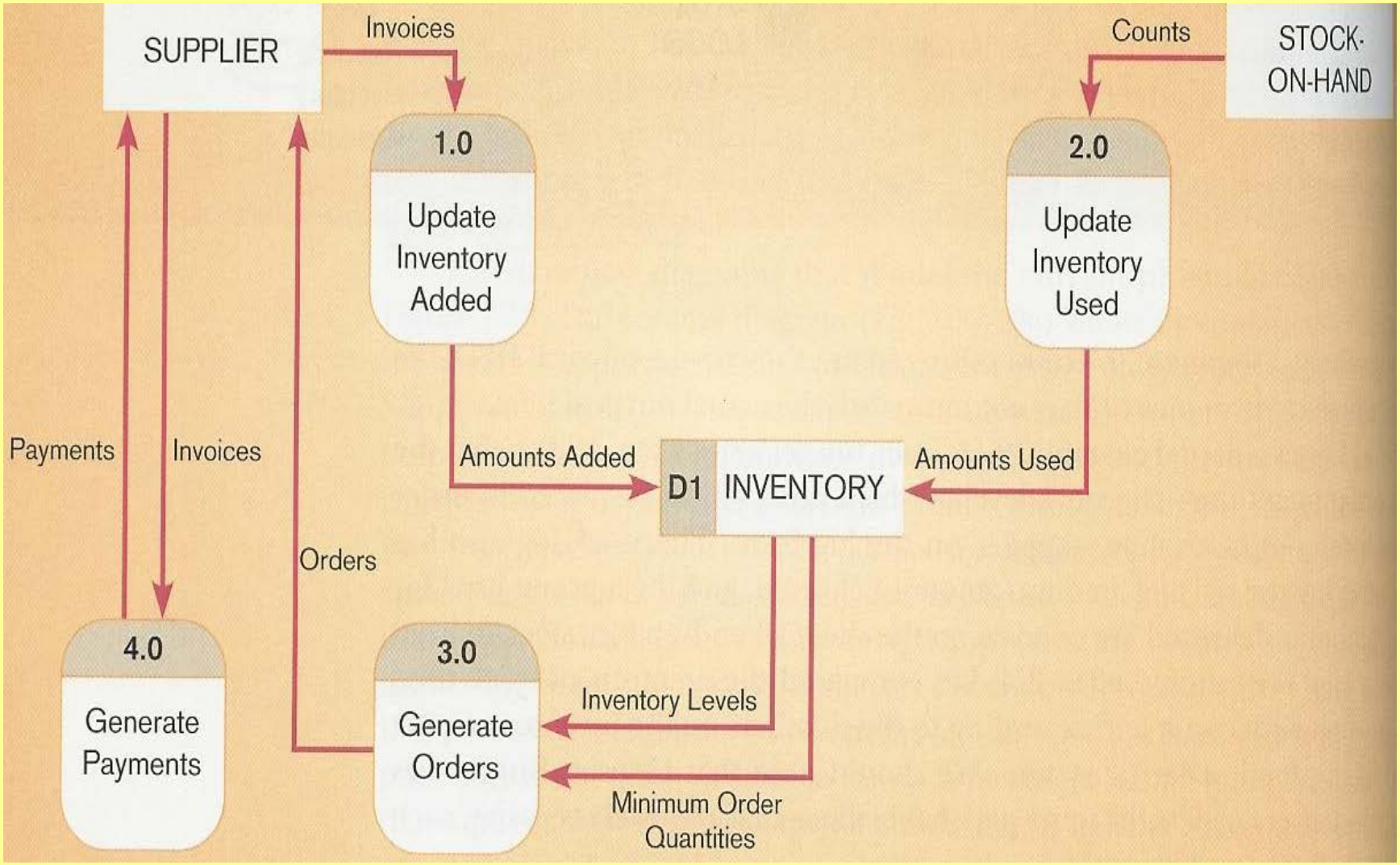
Example: Inventory Control System Contextual Diagram



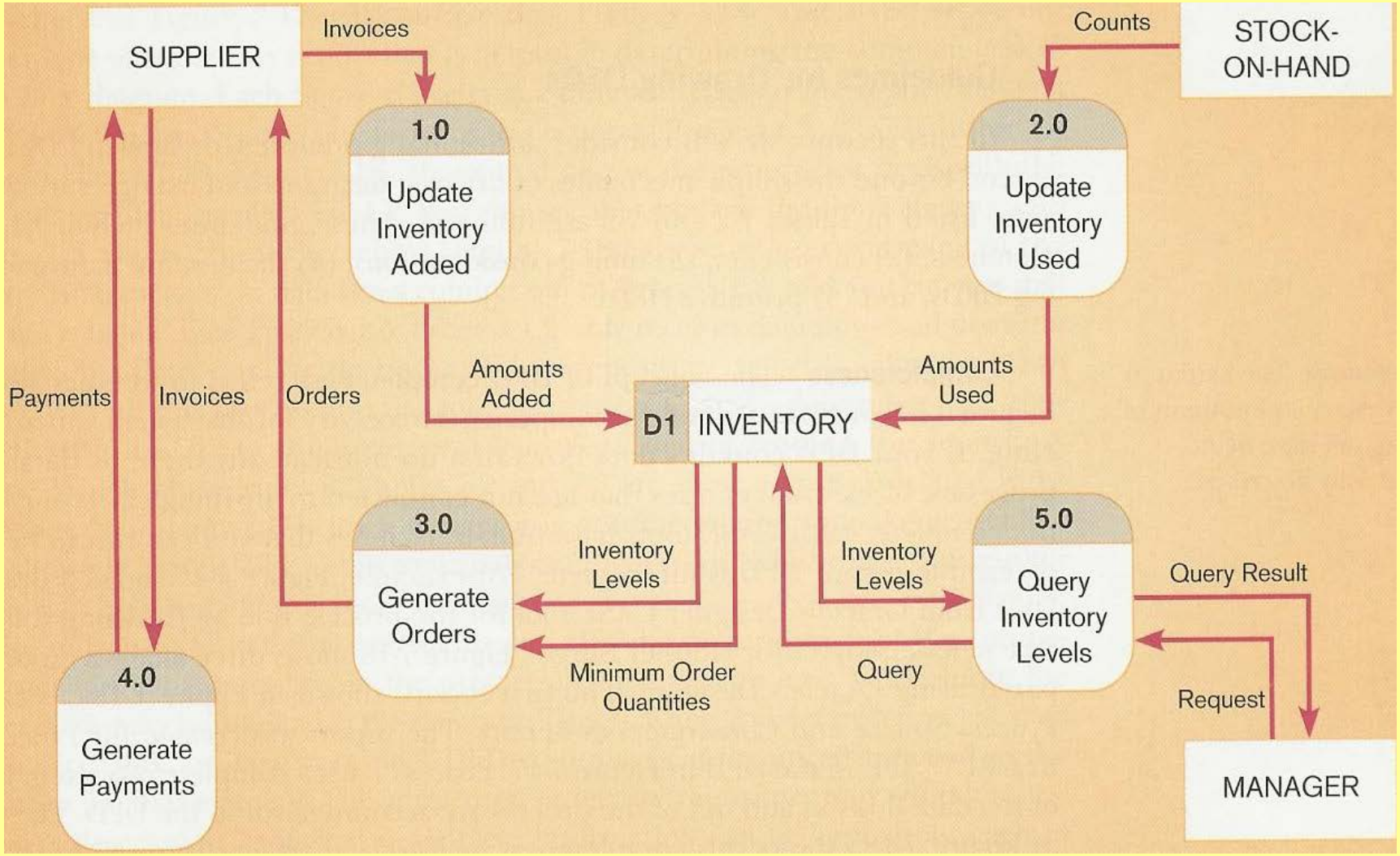
Inventory Control System Current Physical DFD



Inventory Control System Current Logical DFD



Inventory Control System New Logical DFD



Using DFDs as Analysis Tools

- **Gap Analysis**
  - **The process of discovering discrepancies between two or more sets of data flow diagrams or discrepancies within a single DFD**
- **Inefficiencies in a system can often be identified through DFDs**